UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/670,246	09/26/2000	Sujal S. Parikh	3382-55826	5877

7590 07/08/2004

KLARQUIST SPARKMAN CAMPBELL LEIGH & WHINSTON, LLP
One World Trade Center
Suite 1600
121 S. W. Salmon Street
Portland, OR 97204-2988

EXAMINER	
YUAN, ALMARI ROMERO	

ART UNIT	PAPER NUMBER
2176	5

DATE MAILED: 07/08/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/670,246	Applicant(s) PARIKH ET AL.	
	Examiner Almari Yuan	Art Unit 2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 February 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 September 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *4*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to communications: Application filed on 09/26/00 and Information Disclosure Statements filed on 2/05/01.

2. Claims 1-32 are pending in the case. Claims 1, 6, 11, 18, 21, 22, 25, 28, and 29 are independent claims

Information Disclosure Statement

3. The references listed in the Information Disclosure Statements filed on 2/05/01 have been considered.

Claim Objections

4. Claim 11 is objected to because of the following informalities: Claim 11 ends with a semicolon (;), instead of a period (.). Appropriate correction is required.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. **Claims 1-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over "International Layout in CSS", 01/1999, W3C, pages 1-40 (herein after "CSS") in view of Chester et al., "Mastering Excel 97", 1997, Fourth Edition, Sybex, Inc., pages 129-134 (herein after "Excel").**

Regarding independent claim 1 and (dependent claims 8 and 11), CSS discloses:

In a computer system with a textual mark-up language engine, a table sizing method designed to auto-size table elements, a table cell sizing method designed to auto-size table cell elements with a horizontal character flow property, a method for altering the input and output between the table sizing method and the table cell sizing method to render auto-sized textual mark-up language table cells with a vertical character flow property (CSS on page 4-5 and 7 teaches layout flow from horizontal layout to vertical layout; on page 8 teaches document grid used for characters in documents written in East Asian languages (table cell)), the method comprising:

determining that a call to a table cell sizing method from a table sizing method is a request to determine an acceptable minimum width table cell measurement for a given table cell and the request includes a minimum width input designed to be used by a table cell element with the horizontal character flow property to render a minimum table cell width (CSS on page 10 and 14 teaches the width of the grid space is determined by the 'layout-grid-char' setting; on page 15 teaches determining the character grid in a horizontal layout to render within that horizontal space);

determining that the given table cell content element has the vertical character flow property (CSS on page 7 teaches horizontal text in vertical layout determined by the type of character such as East Asian characters; on pages 13-15 teaches vertical layout property);

increasing the minimum width input enough so that it is likely that all characters or objects within each paragraph in the table cell will be flowed into a separate single vertical line (CSS on pages 5 and pages 13-15 shows the characters are rendered in the grid from horizontal flow into the vertical flow of separate single vertical lines);

and calling the table cell sizing method with the increased input (CSS on page 12 teaches setting the line grid value and determining line spacing increment within the grid).

However, CSS does not explicitly disclose "auto-sizing" or "table cell sizing method".

Excel on page 129 teaches the characters within the cell can be modified from horizontal alignment to vertical alignment of text within the cell; also on page 129 item 5.3 and page 134 shows the user can select the orientation of the cells such as rotation of 90 degrees and the row height automatically adjusts when the cell is oriented vertically by increasing the row height to fit the entire entry; on pages 131-133 teaches by selecting vertical alignment the text is automatically adjusted and the height automatically increases to fit multiple lines of text (see page 132).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Excel into CSS to provide a way to automatically modify the size of cells to align text vertically, as taught by Excel, incorporated into the document grid of CSS, in order to enhance the text appearance within a multiple cell environment.

Regarding dependent claim 2, CSS discloses:

where an output from a call to the table cell sizing method consists of a logical height measurement and a logical width measurement, and the logical height measurement is returned

to the table sizing method as the acceptable minimum width for the given table cell (CSS on pages 12-13 teaches determining the logical height and logical width within the line grid).

Regarding dependent claim 3, CSS discloses:

where an output from the call to the table cell sizing method is a measurement of a character or other object in the content element with a longest logical width measurement, the longest logical width measurement being determined by measuring each object or character in the vertical direction (CSS on pages 13-14 teaches determining the line length to adjust the layout length of the characters).

Regarding dependent claims 4 and 16-17, CSS discloses:

where a next call by the table sizing method regarding the given table cell is a request to size that table cell at a proposed width of less than or equal to said logical height returned to the table sizing method as the acceptable minimum table cell width plus some minimal acceptable error, and the logical height and logical width measurements are returned to the table sizing method, the logical height returned as physical width, and the logical width returned as physical height (CSS on pages 13-15 teaches and shows adjusting the grid cell from horizontal layout into vertical layout by modifying the height and width of the grid cell to fit the characters and on page 12 shows mapping between 'layout-flow' and the interpretations of 'layout-grid-line).

Regarding dependent claim 5, CSS discloses:

where the acceptable minimum error ranges from ten to thirty percent of the proposed width (CSS on page 15 teaches setting the character grid into a percentage value).

Regarding independent claim 6 and (dependent claim 9), CSS discloses:

In a computer system with a textual mark-up language engine, a table sizing method designed to auto-size table elements, a table cell sizing method designed to auto-size table cell elements with a horizontal character flow property, a method for altering communications between the table sizing method and the table cell sizing method to render auto-sized textual mark-up language table cells with a vertical character flow property (CSS on page 4-5 and 7 teaches layout flow from horizontal layout to vertical layout; on page 8 teaches document grid used for characters in documents written in East Asian languages (table cell)), the method comprising:

determining that a call to a table cell sizing method from a table sizing method is a request to determine an acceptable maximum width table cell measurement for a given table cell and the request includes a maximum width input designed to be used by a table cell element with the horizontal character flow property to render a maximum table cell width (CSS on page 10 and 14 teaches the width of the grid space is determined by the 'layout-grid-char' setting; on page 15 teaches determining the character grid in a horizontal layout to render within that horizontal space);

determining that the given table cell content element has the vertical character flow property (CSS on page 7 teaches horizontal text in vertical layout determined by the type of character such as East Asian characters; on pages 13-15 teaches vertical layout property);

determining an average character logical width for a language contained in the given table cell (CSS on pages 14-15 teaches logical width for the character within the grid);

determining a maximum distance that characters will be allowed to flow in the vertical character flow direction, the determination being made based on some empirically determined number N multiplied by the average character logical width; calling the table cell sizing method

with the maximum width input changed to the determined distance that characters will be allowed to flow in the character flow direction (CSS on pages 13-15 teaches determining the vertical character flow direction from horizontal flow; determining the logical width (page 15); determining the grid line size of how the characters will be flowed).

However, CSS does not explicitly disclose "auto-sizing" or "table cell sizing method".

Excel on page 129 teaches the characters within the cell can be modified from horizontal alignment to vertical alignment of text within the cell; also on page 129 item 5.3 and page 134 shows the user can select the orientation of the cells such as rotation of 90 degrees and the row height automatically adjusts when the cell is oriented vertically by increasing the row height to fit the entire entry; on pages 131-133 teaches by selecting vertical alignment the text is automatically adjusted and the height automatically increases to fit multiple lines of text (see page 132).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Excel into CSS to provide a way to automatically modify the size of cells to align text vertically, as taught by Excel, incorporated into the document grid of CSS, in order to enhance the text appearance within a multiple cell environment.

Regarding dependent claim 7, CSS discloses:

where an output from the call to the table cell sizing method consists of logical height and logical width measurements and the logical height measurement is returned to the table sizing method as the acceptable maximum width table cell (CSS on pages 12-13 teaches determining the logical height and logical width within the line grid).

However, CSS does not explicitly disclose "table cell sizing method".

Excel on page 129 item 5.3 and page 134 shows the user can select the orientation of the cells such as rotation of 90 degrees and the row height automatically adjusts when the cell is oriented vertically by increasing the row height to fit the entire entry; on pages 131-133 teaches by selecting vertical alignment the text is automatically adjusted and the height automatically increases to fit multiple lines of text (see page 132).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Excel into CSS to provide a way to automatically modify the size of cells to align text vertically, as taught by Excel, incorporated into the document grid of CSS, in order to enhance the text appearance within a multiple cell environment.

Regarding dependent claim 10, CSS discloses:

where a next call by the table sizing method regarding the given table cell is a request to size that table cell at a proposed physical width greater than or equal to the logical height returned to the table sizing method as the maximum acceptable width table cell, and the logical height and logical width measurements are returned to the table cell sizing method, the logical height returned as physical width, and the logical width returned as physical height (CSS on pages 13-15 teaches and shows adjusting the grid cell from horizontal layout into vertical layout by modifying the height and width of the grid cell to fit the characters and on page 12 shows mapping between 'layout-flow' and the interpretations of 'layout-grid-line)

Regarding dependent claims 12-14, CSS discloses:

where the estimated logical width input is determined by dividing the area of a minimum and maximum logical height rectangle by a physical width proposed by a table sizing method

(CSS on page 13 teaches determining size of the line grid's unit space relative to the logical height and on page 15 teaches determining logical width).

Regarding dependent claim 15, CSS discloses:

where the estimated logical width input is determined by summing the areas of a minimum logical height rectangle with a maximum logical height rectangle, and dividing that sum by twice a physical width proposed by the table sizing method (CSS on page 15 teaches determining the logical width).

Regarding independent claim 18 and (dependent claim 20), CSS discloses:

In a computer system with a textual mark-up language engine, a parent textual mark-up language element with a horizontal character flow property, a child textual mark-up language element with a vertical character flow property, a method for determining a proposed logical width dimension, the method being given a proposed physical width, a desired layout area (CSS on page 4-5 and 7 teaches layout flow from horizontal layout to vertical layout; on page 8 teaches document grid used for characters in documents written in East Asian languages (table cell)) , and the child element, the method comprising:

determining an area of a minimum logical height rectangle for the child element by flowing each paragraph or sentence in the child element into a single line in the character flow direction (CSS on page 13 teaches determining the logical height from a horizontally layout flow); and

obtaining an actual logical width of a minimum logical height rectangle by measuring a length of a longest line in the child element in the character flow direction (CSS on page 13 teaches the grid line size is determined by the largest character in the element's font);

obtaining an actual logical height of a minimum logical height rectangle by finding and summing the measurements of each character or object in each said line that that occupies the most space in the direction perpendicular to the character flow direction and adding a minimal space between lines to improve readability (CSS on pages 12-14 teaches based on the 'layout-grid-line which measures the grid size and grid height when in a vertical layout which affects the dimension in a perpendicular direction);

and multiplying the obtained actual logical height by the obtained actual logical width; determining an area of a maximum logical height rectangle for the child element by (CSS on page 13 teaches determining logical height to obtained logical width on page 15);

setting a maximum logical width for the maximum logical height rectangle (CSS on page 12 teaches property sets the line grid value for an element which can be determining logical width and logical height);

flowing the child element into the maximum logical height rectangle in a line in the character flow direction so long as no next character or object in the content element being flowed into the line would cause the length of the line to exceed the set maximum logical width (CSS on pages 12-15 teaches the grid is adjusted based on the direction of the character layout flow from horizontal into vertical);

starting a new line next to and parallel to the previous line and flowing the content element into that line each time the next character or object in the child element would cause the line then being flowed into the rectangular area to exceed the set maximum logical width (CSS on pages 14-15 teaches vertical layout with lines parallel to one another);

obtaining an actual logical width of the maximum logical height rectangle by finding and measuring a length of the longest line in the character flow direction (CSS on page 13 teaches the grid line size is determined by the largest character in the element's font);

obtaining an actual logical height of the minimum logical height rectangle by finding and summing the measurements of each character or object in each said line that occupies the most space in the direction perpendicular to the character flow direction and adding a minimal space between lines to improve readability (CSS on pages 12-14 teaches based on the 'layout-grid-line which measures the grid size and grid height when in a vertical layout which affects the dimension in a perpendicular direction);

multiplying the obtained actual logical height by the obtained actual logical width in order to determine the area of the maximum logical height rectangle (CSS on page 13 teaches determining logical height to obtained logical width on page 15);

determining a proposed logical width by summing the areas of the determined maximum logical height rectangle and the determined minimum logical height rectangle and dividing the sum by approximately twice the proposed physical width (CSS on page 15 teaches determining the logical width).

However, CSS does not explicitly disclose "auto-sizing" or "table cell sizing method".

Excel on page 129 teaches the characters within the cell can be modified from horizontal alignment to vertical alignment of text within the cell; also on page 129 item 5.3 and page 134 shows the user can select the orientation of the cells such as rotation of 90 degrees and the row height automatically adjusts when the cell is oriented vertically by increasing the row height to fit the entire entry; on pages 131-133 teaches by selecting vertical alignment the text is

automatically adjusted and the height automatically increases to fit multiple lines of text (see page 132).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Excel into CSS to provide a way to automatically modify the size of cells to align text vertically, as taught by Excel, incorporated into the document grid of CSS, in order to enhance the text appearance within a multiple cell environment.

Regarding dependent claim 19, CSS discloses:

where the determined proposed logical width is set at the greater of itself or the value determined by dividing the area of the determined minimum logical height rectangle area by the proposed physical width (CSS on page 13 teaches determining size of the line grid's unit space relative to the logical height and on page 15 teaches determining logical width).

Regarding dependent claim 26, CSS discloses:

rotated ninety degrees clockwise (CSS on page 5 teaches characters rotated 90 degrees clockwise).

Regarding claims 21-25 and 27-32, the limitations of claims 21-25 and 27-32 incorporates similar subject matter as claimed in claims 1-20 and 26, and are rejected along the same rationale.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Almari Yuan whose telephone number is 703-305-5945. The examiner can normally be reached on Mondays - Fridays (8:30am - 5:00pm).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Joseph Feild, can be reached on 703-305-9792. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AY
June 22, 2004

JOSEPH FEILD
SUPERVISORY PATENT EXAMINER